



Configuring two-way SSL using Estonian ID-cards in the Ubuntu Apache2 web server

Document information	
Date of creation	06/02/2019
Client	information System Authority
Author	Urmas Vanem, OctoX
Version	25.10/1

Version information		
Date	Version	Changes/notes
06/02/2019	19.02/1	Public version
20/02/2019	19.02/1	Added the chapter of additional configuration options: firewall and OCSP configuration, default website removal. Changed by Urmas Vanem
12/12/2019	19.12/1	Added recommendations for securing Apache. Changed by Urmas Vanem
16/12/2020	20.12/1	Added a requirement for the user certificate to have the correct <i>extendedKeyUsage</i> field and the right certificate issuer. See the chapter 'Additional filtering of user certificates'. Changed by Urmas Vanem
17/12/2020	20.12/2	Added the directive SSLCADNRequestPath, see the chapter 'Filtering certificates displayed to the user'. Changed by Urmas Vanem
13/01/2021	21.01/1	Added the demonstrative configuration file as a link. Added HSTS configuration. Changed by Urmas Vanem
21/01/2021	21.01/2	SSLOCSPEnable directive replaced from <i>on</i> to <i>leaf</i> . Updated TLS 1.2 cipher recommendations and TLS protocol usage recommendations. Variable names in Democonf and document have been synchronised. Changed by Urmas Vanem
27/01/2021	21.01/3	Added the mobile-ID filter. Changed by Urmas Vanem

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards

26/02/2021	21.02/1	Added the alternative possibility to filter intermediate certificate authorities using the SSLCADNRequestFile directive. Changed by Urmas Vanem
27/04/2021	21.04/1	Support for outdated ESTEID-SK 2011 certificates removed. Changed by Urmas Vanem
25/11/2021	21.11/1	Ubuntu version updated to Ubuntu Server 21.10. Apache version updated to 2.4.48. Added guidance for ECC certificates. Updated TLS and cipher recommendations.
21/02/2023	23.02/1	Ubuntu version updated to Ubuntu Server 22.04. Apache version updated to 2.4.55. Updates in the virtual host configuration. Changed by Urmas Vanem
27/12/2023	23.12/1	Removed the ESTEID-SK 2015 chain. Changed by Urmas Vanem
27/12/2023	23.12/2	Removed the outdated OCSP responder certificate. Changed by Urmas Vanem
22/08/2024	24.08/1	Ubuntu version updated to Ubuntu Server 24.04. Apache version updated to 2.4.62. Updates in the virtual host configuration. Changed by Urmas Vanem
31/10/2025	25.10/1	Added Zetes certificates Changed by Raul Kaidro



Introduction

This guide describes:

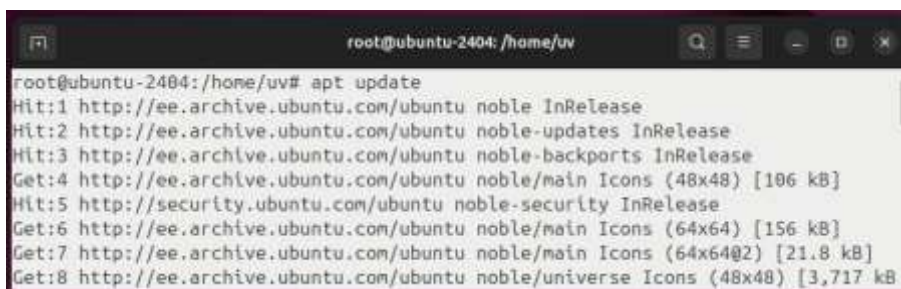
- How to install and configure the Apache2 (v. 2.4.65) web server on Ubuntu Server 24.04.
- How to configure HTTPS (one-way SSL) in the web server.
- How to configure ID-card authentication (two-way SSL) in the web server using ID-cards.
- Other options for server configuration and recommendations for ensuring security.

Apache2 installation and configuration

Installation

1. Renew the Ubuntu package data – in the terminal, run

apt update

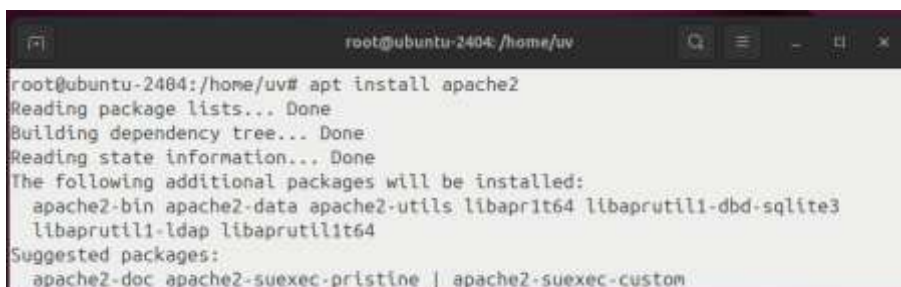


```
root@ubuntu-2404: /home/uv# apt update
Hit:1 http://ee.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ee.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ee.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://ee.archive.ubuntu.com/ubuntu noble/main Icons (48x48) [106 kB]
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Get:6 http://ee.archive.ubuntu.com/ubuntu noble/main Icons (64x64) [156 kB]
Get:7 http://ee.archive.ubuntu.com/ubuntu noble/main Icons (64x64@2) [21.8 kB]
Get:8 http://ee.archive.ubuntu.com/ubuntu noble/universe Icons (48x48) [3,717 kB]
```

Picture 1 – updating packages

2. Install Apache2 with the command

apt install apache2



```
root@ubuntu-2404: /home/uv# apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libaprutil1t64
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
```

Picture 2 – Apache2 installation



Ubuntu/Apache2 SSL configuration

A simple configuration guide for Estonian ID-cards

As result of the previous steps, the Apache server is now installed¹.

```
parallels@ubuntu-linux-24-04-desktop:~$ apache2 -v
Server version: Apache/2.4.58 (Ubuntu)
Server built:   2025-08-11T11:10:09
parallels@ubuntu-linux-24-04-desktop:~$
```

Picture 3 – Apache version query

Update Apache to version 2.4.65 using the following commands:

```
add-apt-repository ppa:ondrej/apache2
```

```
apt update
```

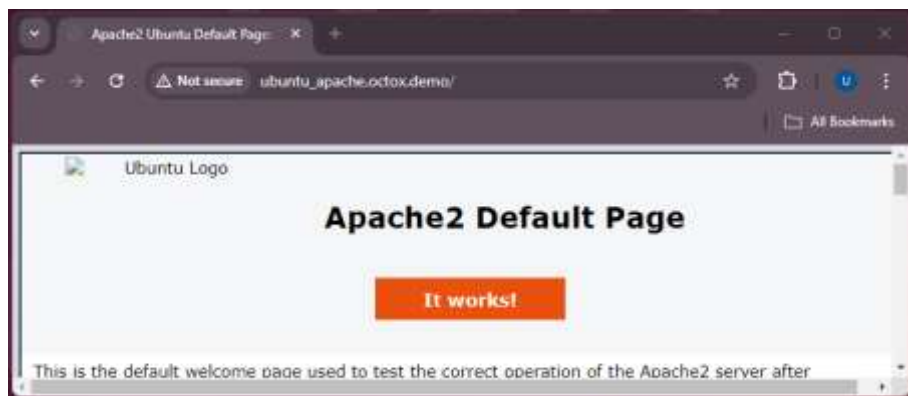
```
apt upgrade
```

Apache has now been successfully updated to version 2.4.65 as expected:

```
parallels@ubuntu-linux-24-04-desktop:~$ apache2 -v
Server version: Apache/2.4.65 (Ubuntu)
Server built:   2025-07-26T17:41:22
parallels@ubuntu-linux-24-04-desktop:~$
```

Picture 4 – the updated Apache version is 2.4.65

With version 2.4.65, the Apache2 web server runs in the insecure HTTP mode:



Picture 5 – Apache web server in the default configuration

Configuration

Enabling one-way SSL

Enable SSL for Apache2 with the command

```
a2enmod ssl
```

¹ Currently (as at 30/10/2025), Apache version 2.4.58 is included with Ubuntu by default. The latest version of Apache is 2.4.65.



and restart the Apache2 service

```
systemctl restart apache2
```

```
root@ubuntu-2404:/home/uv# a2enmod ssl
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create s
elf-signed certificates.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@ubuntu-2404:/home/uv# systemctl restart apache2
root@ubuntu-2404:/home/uv#
```

Picture 6 – enabling SSL and restarting the service

Creating the private key and the Certification Signing Request (CSR) file

Elliptic Curve Cryptography (ECC)

First, generate an ECC private key:

```
openssl ecparam -name secp384r1 -genkey -noout -out Apache2404.key
```

Then, generate an ECC CSR:

```
openssl req -new -key Apache2404.key -out Apache2404.csr -subj
/C=EE/O=OctoX/CN=Apache2404.octox.demo -reqexts SAN -config <(cat
/etc/ssl/openssl.cnf <(printf
"[SAN]\nsubjectAltName=DNS:Apache2404.octox.demo,DNS:MYWEB
SERVER.octox.demo"))2
```

```
root@ubuntu-2404:/home/uv# openssl ecparam -name secp384r1 -genkey -noout -out A
pache2404.key
root@ubuntu-2404:/home/uv# openssl req -new -key Apache2404.key -out Apache2404.
csr -subj /C=EE/O=OctoX/CN=Apache2404.octox.demo -reqexts SAN -config <(cat /etc
/ssl/openssl.cnf <(printf "[SAN]\nsubjectAltName=DNS:Apache2404.octox.demo,DNS:M
YWEBSERVER.octox.demo"))
root@ubuntu-2404:/home/uv#
```

Picture 7 – creating a private key and a CSR

Parameters on the yellow background:

1. Apache2404.key is the private key of the certificate;
2. Apache2404.csr is the CSR for the certificate authority (CA);
3. CN=Apache2404.octox.demo is the common name for the certificate;

² In addition to the certificate attributes C, O, and CN described on the command line, it is also possible to describe the attributes L, OU, and S if desired. However, only CN can also be used.



4. DNS:Apache2404.octox.demo and DNS:MYWEB SERVER.octox.demo are the SAN DNS names for the certificate. These names must correspond to the actual address of the website³. The names must also be resolvable in name services.

The contents of the CSR can be viewed by running

```
openssl req -in Apache2404.csr -noout -text
```

```
root@ubuntu-2404:/home/uv# openssl req -in Apache2404.csr -noout -text
Certificate Request:
  Data:
    Version: 1 (0x0)
    Subject: C = EE, D = OctoX, CN = Apache2404.octox.demo
    Subject Public Key Info:
      Public Key Algorithm: id-ecPublicKey
      Public-Key: (384 bit)
      pub:
        04:db:9b:fe:8c:11:87:00:b1:71:9b:54:06:3a:49:
        71:b0:89:04:dc:a9:75:52:54:42:39:07:21:84:51:
        b7:5b:07:61:09:5b:e7:82:ff:60:58:b3:af:5e:73:
        ee:03:47:1d:9d:26:e6:fe:92:e0:60:df:71:23:8e:
        24:2b:11:be:68:f6:60:6c:3e:be:dc:7d:f4:32:6e:
        9e:ae:5e:73:5f:fd:43:74:ab:8d:7d:d8:91:b6:e1:
        52:f9:f6:53:aa:df:64
      ASN1 OID: secp384r1
      NIST CURVE: P-384
    Attributes:
      Requested Extensions:
        X509v3 Subject Alternative Name:
          DNS:Apache2404.octox.demo, DNS:MYWEBSESERVER.octox.demo
    Signature Algorithm: ecdsa-with-SHA256
    Signature Value:
```

Picture 8 – the contents of the CSR

RSA

The following section on creating RSA CSR is left here from the previous version of the document if somebody wants to continue using certificates based on the RSA algorithm. In the following chapters of the document, certificates based on the ECC algorithm are described.

Create a CSR and a private key with the command

```
openssl req -newkey rsa:2048 -keyout Apache2021.key -sha256 -subj
"/CN=Apache5.kaheksa.xi" -reqexts SAN -config <(cat
/etc/ssl/openssl.cnf <(printf
"[SAN]\nsubjectAltName=DNS:Apache2021.kaheksa.xi,DNS:Apache5.kaheksa
.xi")) -out Apache2021.csr -nodes
```

³ Modern browsers do not trust websites where at least one SAN DNS name is not equal to the actual address of the website.



```
uv@Ubuntu8: ~  
File Edit View Search Terminal Help  
uv@Ubuntu8:~$ openssl req -newkey rsa:2048 -keyout Apache2021.key -sha256 -subj  
"/CN=Apache5.kaheksa.xi" -reqexts SAN -config <(cat /etc/ssl/openssl.cnf <(print  
f "[SAN]\nsubjectAltName=DNS:Apache2021.kaheksa.xi,DNS:Apache5.kaheksa.xi")) -ou  
t Apache2021.csr -nodes  
Generating a RSA private key  
.....+++++  
.+++++  
writing new private key to 'Apache2021.key'  
-----  
uv@Ubuntu8:~$
```

Picture 9 – generating the private key and CSR

Parameters on the yellow background:

1. Apache2021.key is the private key of the certificate;
2. Apache2021.csr is the CSR for the CA;
3. Apache5.kaheksa.xi is the subject name for the certificate;
4. Apache2021.kaheksa.xi and Apache5.kaheksa.xi are the SAN DNS names for the certificate. These names must correspond to the actual address of the website⁴. The names must also be resolvable in name services.

The contents of the CSR can be viewed with the command

```
openssl req -in Apache2021.csr -noout -text
```

⁴ Modern browsers do not trust websites where at least one SAN DNS name is not equal to the actual address of the website.

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards

```
uv@Ubuntu8: ~
File Edit View Search Terminal Help
uv@Ubuntu8:~$ openssl req -in Apache2021.csr -noout -text
Certificate Request:
Data:
  Version: 1 (0x0)
  Subject: CN = Apache5.kaheksa.xl
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public-Key: (2048 bit)
    Modulus:
      00:c9:4f:a2:54:bd:1a:bb:88:a6:ec:16:c9:3e:28:
      ee:f6:f6:09:3a:d7:e6:8f:a6:7a:4e:57:3b:38:70:
      70:73:b0:01:95:7a:8d:c3:47:46:49:b9:12:52:20:
      08:0c:ed:f5:ec:c5:4e:25:3e:27:9b:98:67:b0:bd:
      c2:cd:00:98:54:36:d4:bf:b8:60:d9:aa:26:de:6a:
      da:11:23:2e:a9:05:94:ff:e8:bb:d2:5e:c2:68:8d:
      63:97:71:5e:0a:a0:49:fc:27:c7:28:c4:7d:53:12:
      1c:e6:2e:9d:bd:81:5b:ff:6a:e5:cf:b5:1a:1b:a3:
      5a:2e:9b:bd:0c:fe:c8:8f:ed:ff:b0:08:9a:1a:69:
      4f:88:a1:1c:c7:9d:84:53:f0:77:2f:db:ba:2a:9a:
      16:f4:78:02:ca:e2:29:f7:f0:f3:61:df:00:ce:3f:
      fa:80:c5:ca:2d:37:a4:2e:a4:8c:be:a2:b3:c9:fd:
      46:4e:20:fb:18:8b:3d:09:6a:be:01:3d:af:29:dd:
      e2:b6:63:3c:3e:46:c1:7a:9b:08:83:c9:32:c5:54:
      b2:e6:3d:a3:68:b6:8d:53:cb:36:c2:20:7d:77:63:
      c7:cf:c9:11:36:b3:47:9b:10:8f:19:66:cb:a4:0f:
      50:f5:35:bf:0d:53:82:cb:ad:3c:1f:5a:1a:2b:70:
      a4:8f
    Exponent: 65537 (0x10001)
  Attributes:
  Requested Extensions:
    X.509v3 Subject Alternative Name:
      DNS:Apache2021.kaheksa.xl, DNS:Apache5.kaheksa.xl
  Signature Algorithm: sha256WithRSAEncryption
```

Picture 10 – the contents of the CSR

Ordering and installing an SSL certificate

The CSR **Apache2404.csr** should be sent to trustworthy CA. In the demo environment, the certificate issuer is the test CA. Signed certificate is issued in Base64 encoded format.

```
-----BEGIN CERTIFICATE-----
MIICGDCCA26gAwIBAgITeQAAAAFueX80MmM5gAAAAAACTAKBggqhkiJOPQ00AzaR
MQBwDQYDVQQDEwZka5hbeUwhcNMhjqv00E4NTQWtzU34hcNMhjqv00E4MTQ1MzU3
KjA9MQswCQYDVQQGEwJFRTEOMAwGA1UECHFTZm81gXhHAcBgNVBAMTFUwYwWlo
ZTI0MQswC2Nhb3ZGVtYzB2MBAGByqGSM49AgEGSUsBAAIAZ1ABNub/owRhwCx
c2tU8jppJcbCJ8HypdVJUQjkhIYRRt1aHYQ1b54L/YF1zr1Sz7gMH20w5v6S46Df
c500jCsRvzj2Y0wvrtx990juna5ec1/9Q35rjX3YkbbhJm2U6+fZK0B1zCB10A4
BgNVHREEMTAvghV8cGFJaGJyNDAA0Lw9jaG94LmR1bW+Cfk1ZV8VCUBMSVwVUw9j
dG94LmR1bW8wHQYDVRR0088YEFAsQZ0UgEIJvGMVZ7qthYV50cyIMB8GA1UdIwQY
MBaAFHwUhsN1VKVBo6FY8ap5K16M+9UTHAwGA1UdEwEB/wQMAAwGyIkoZ1zj0E
AwMDaAAwZQIwtdm+C8whTsYVnZ4axxQhtAx5ak9/kZGctpMe32x2kyjE1Zow+RIyn
cuFwt1CgYIwLAjEAgj69bKlQ50MQ=4FRVMr1V56fp6FjPH7qsh5U5UgYnFAL5iU
o80umYynxvZsuxES
-----END CERTIFICATE-----
```

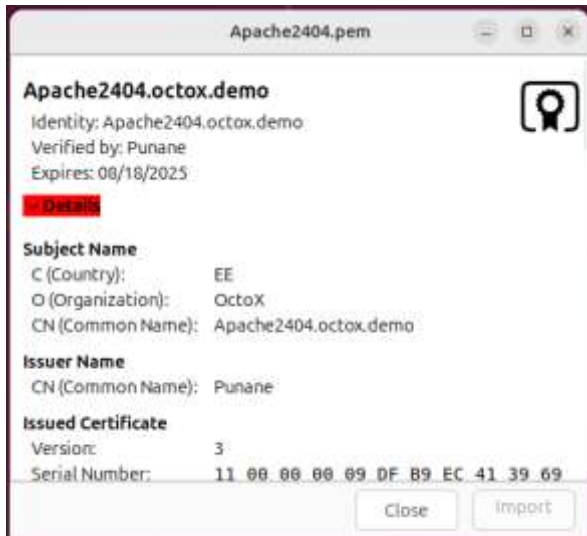
Picture 11 – Base64 format certificate in a text redactor

In Ubuntu, the certificate looks like the following picture:

Ubuntu/Apache2 SSL configuration

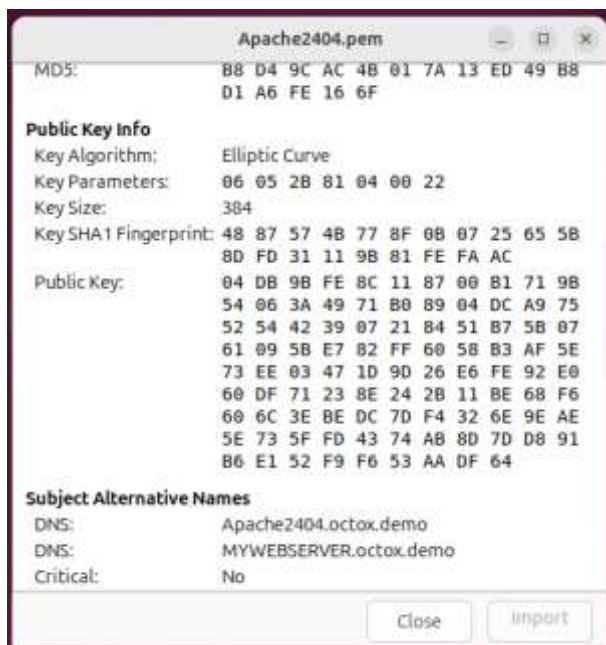


A simple configuration guide for Estonian ID-cards



Picture 12 – ECC certificate in Ubuntu

The certificate also includes the algorithm and alternative SAN DNS names of the subject:



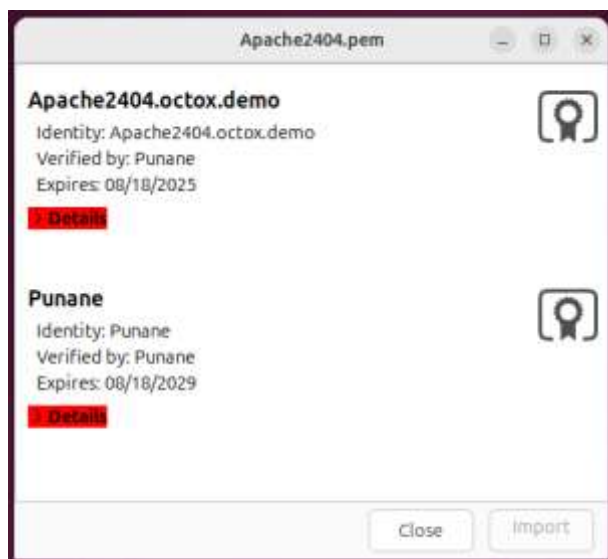
Picture 13 – the algorithm and SAN DNS names

As you can see, the certificate issuer is a CA named 'Punane'. Now, you need to create a certificate file in which both the TLS certificate of the future web server and its chain of issuers are located. To do this, add the issuer's certificate in .pem format to the certificate file of the web server in .pem format and save the file as **Apache2404.pem**.

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards



Picture 14 – the certificates are consolidated into a single file

Place the generated file in the `/etc/ssl/certs` folder. In addition, you need to place the certificate private key in the `/etc/ssl/private` folder.



Picture 15 – copying the certificate and its private key to the certificates folder

Now, the certificates and private key needed by Apache2 for one-way SSL have been correctly installed.

Creating a virtual website

Create a separate virtual website for your configuration. First, create a home folder named `/var/www/Apache2404` for the content of the website.



Picture 16 – creating a home folder for the website

Place a simple and recognisable webpage in the folder. In this example, the file `/var/www/html/index.html` is copied to the new folder for testing. Minor modifications are made in the heading or title of the copied webpage to ensure it is taken from the right place.

Next, prepare the virtual site configuration file. Create a new file named with the command

```
nano /etc/apache2/sites-available/Apache2404.conf
```

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards

```
root@ubuntu-2404: /home/uv
root@ubuntu-2404: /home/uv# nano /etc/apache2/sites-available/Apache2404.conf
root@ubuntu-2404: /home/uv#
```

Picture 17 – creating a new configuration file

Now, change the new configuration file as you wish. Paste the following configuration in it:

```
# Beginning of file

<Virtualhost Apache2404.octox.demo:80>

# By contacting the HTTP site, automatic HTTP -> HTTPS redirection takes
place with the next two lines.
    Servername Apache2404.octox.demo
    redirect / https://Apache2404.octox.demo
</Virtualhost>

<VirtualHost Apache2404.octox.demo:443>

# General info
    ServerName Apache2404.octox.demo:443
    DocumentRoot /var/www/Apache2404

# SSL configuration
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/Apache2404.pem
    SSLCertificateKeyFile /etc/ssl/private/Apache2404.key

# Error collection configuration
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</Virtualhost>

# End of file
```

Activate the new configuration with the command

a2ensite Apache2404.conf

Then, restart the Apache2 service.

```
root@ubuntu-2404: /home/uv
root@ubuntu-2404: /home/uv# a2ensite Apache2404.conf

Enabling site Apache2404.
To activate the new configuration, you need to run:
  systemctl reload apache2
root@ubuntu-2404: /home/uv# systemctl reload apache2
root@ubuntu-2404: /home/uv#
```

Picture 18 – activating the website and restarting Apache2

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards

Now, the new website can be accessed by one-way SSL. In addition, all HTTP requests to the site <http://Apache2404.octox.demo> are automatically redirected to the HTTPS site <https://Apache2404.octox.demo>.



Picture 19 – the Apache web server is working and using one-way SSL

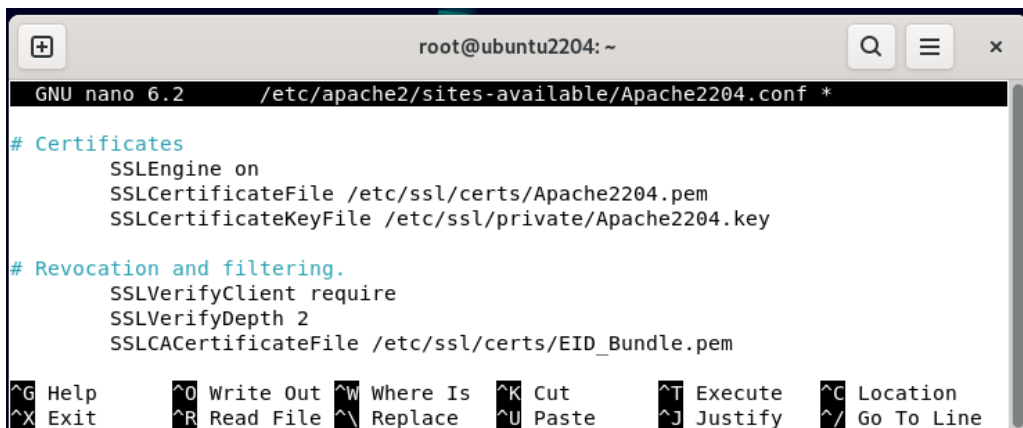
Note. There can be many similar virtual websites with different names in same Apache2 server with a single IP address.

Requiring two-way SSL

If you wish to allow website access by authenticating with an Estonian ID-card, you need to supplement the existing configuration slightly.

Add the following lines to the SSL section of the file Apache2404.conf:

```
SSLVerifyClient                                require
SSLVerifyDepth                                2
SSLCACertificateFile /etc/ssl/certs/EID_Bundle.pem
```



Picture 20 – updated configuration file, SSL section

Now, create a new text file named **EID_Bundle.pem**⁵, which includes the eID root and intermediate certificates (EE-GovCA2018, ESTEID2018, EEGovCA2025, ESTEID2025) in Base64 encoded format. With this file, you can filter out all CA's whose certificates are supported by the new website. The user will only see the certificates from those chains. When opened in Ubuntu, the file looks like this:

⁵ Available: EID_Bundle.pem

Ubuntu/Apache2 SSL configuration



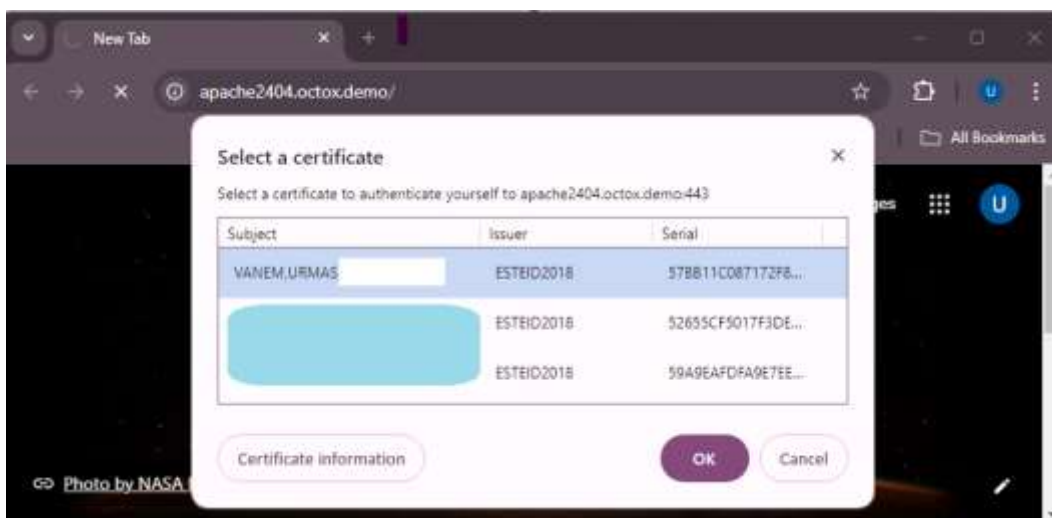
A simple configuration guide for Estonian ID-cards



Picture 21 – root and intermediate certificates in one file

Save the file as **EID_Bundle.pem** and copy it to the folder **/etc/ssl/certs**. Restart Apache2 web server with the command `'systemctl reload apache2'` to activate the change in the web server.

After accessing the website `Apache2404.octox.demo` now, a user certificate is required.



Picture 22 – user certification request

The server suggests certificates to the user, the issuers of which are described in the file **EID_Bundle.pem**. After confirming the certificate and entering the PIN, the user can access the website – two-way SSL works.



Additional configuration options

The purpose of this document is not to give exact guidance on how to optimise or protect websites, but to show how to configure two-way SSL for Estonian ID-cards. However, you should take into account the following.

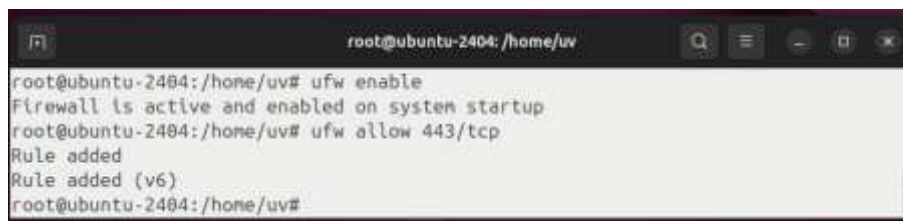
Firewall rules (if necessary)

For creating a firewall rule, run the command

```
ufw allow 'DESIRABLE RULE'
```

on the terminal. For example, to allow HTTPS traffic only, run

```
ufw allow 443/tcp
```



```
root@ubuntu-2404: /home/uv
root@ubuntu-2404: /home/uv# ufw enable
Firewall is active and enabled on system startup
root@ubuntu-2404: /home/uv# ufw allow 443/tcp
Rule added
Rule added (v6)
root@ubuntu-2404: /home/uv#
```

Picture 23 – activating firewall and creating a https rule

If the firewall is active (**ufw enable**), running the command **ufw status** in the terminal shows the active rules.



```
root@ubuntu-2404: /home/uv# ufw status
Status: active

To Action From
--
443/tcp ALLOW Anywhere
443/tcp (v6) ALLOW Anywhere (v6)

root@ubuntu-2404: /home/uv#
```

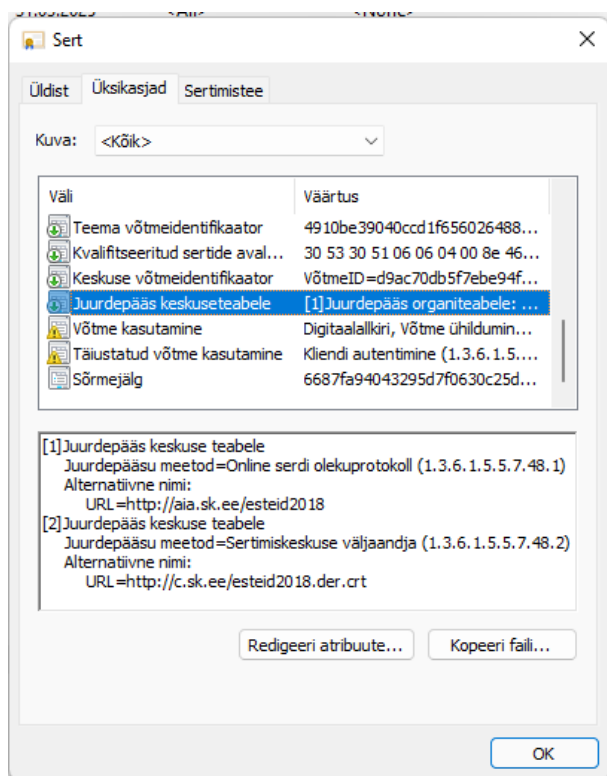
Picture 24 – the firewall is active and only HTTPS traffic is enabled



Checking the status of the user's certificate against the OCSP service⁶

Using the OCSP (Online Certificate Status Protocol) service, you can check the revocation status of client certificates practically in real time. With every client authentication attempt, the web server sends a query to the OCSP service, which responds with the certificate status.

SK and Zetes offer a free-access (free-of-charge) AIA OCSP service. For certificates issued under the ESTEID2018 and ESTEID2025 CA, AIA OCSP service location is already included in the certificate (<http://aia.sk.ee/esteid2018>, <http://ocsp.eidpki.ee>).



Picture 25 – ESTEID2018 AIA OCSP address in the certificate

To enable the user certificate validity check against the AIA OCSP service, you need to add the following lines to the SSL configuration of Apache2:

```
SSLCOSEnable leaf # - enables the OCSP validity check for user certificates.  
SSLCOSEnableRequestNonce off # - turns off the OCSP service nonce requirement.
```

⁶ The certificate check is also doable with certificate revocation lists (CRL), but this is not covered in this document, as the OCSP-based solution is preferred.



Ubuntu/Apache2 SSL configuration

A simple configuration guide for Estonian ID-cards

```
root@ubuntu2204: ~
GNU nano 6.2 /etc/apache2/sites-available/Apache2204.conf
# Certificates
  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/Apache2204.pem
  SSLCertificateKeyFile /etc/ssl/private/Apache2204.key

# Revocation and filtering.
  SSLVerifyClient require
  SSLVerifyDepth 2
  SSLCACertificateFile /etc/ssl/certs/EID_Bundle.pem

# AIA-OCSP
  SSLOCSPEnable leaf
  SSLOCSPUseRequestNonce off

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^N Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Picture 26 – the AIA OCSP configuration is added

Reload the Apache2 web service with the command `systemctl reload Apache2`. With this configuration, the OSCP service address is taken from the user certificate.

Default webpage removal

The default webpage is also installed with the Apache2 installation. To remove the default website from the solution, run the command `a2dissite 000-default.conf` and activate the change with `systemctl reload apache2`.

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
systemctl reload apache2
root@ubuntu2204: /home/uv/temp# systemctl reload apache2
root@ubuntu2204: /home/uv/temp#
```

Picture 27 – default webpage removal

Recommended security settings for Apache

SSL/TLS

Apache version 2.4.55 uses all SSL/TLS protocols with a version higher than SSL3 by default:

```
root@ubuntu2204: /home/uv/temp
root@ubuntu2204: /home/uv/temp# grep -i -r "SSLProtocol" /etc/apache2/nods-availa
ble/
/etc/apache2/nods-available/ssl.conf:SSLProtocol all -SSLv3
root@ubuntu2204: /home/uv/temp#
```

Picture 28 – default configuration of Apache SSL/TLS

It is no longer recommended to use TLS protocols with a version number lower than TLS 1.2. TLS version 1.3 has also been in use for a while.



If there is no specific requirement to allow TLS 1.2, it is recommended to only use TLS 1.3. While TLS 1.2 is very stable and secure with the correct configuration, TLS version 1.3 is faster, more secure by default, and needs less configuration. In standard situations, TLS 1.2 should be enabled only if really needed, and if it is enabled, it is mandatory to allow only secure cipher suites and extensions.

To configure Apache to support only TLS protocol version 1.3, you need to add the following line to the Apache configuration file:

```
SSLPROTOCOL -all +TLSv1.3
```

```
SSLPROTOCOL -all +TLSv1.3
```

Picture 290 – enabling only TLS version 1.3 in the configuration file

To support TLS versions 1.2 and 1.3, add **+TLSv1.2** to the configuration line.

If you want to make the change at the server level, modify the parameter SSLPROTOCOL in the file `/etc/apache2/mods-available/ssl.conf`.

More information about the recommendations for the use of the cipher suites can be found in the cryptographic algorithms life cycle report ordered by RIA at <https://www.id.ee/en/article/cryptographic-algorithms-life-cycle-reports-2/>.

Cipher suites

All TLS 1.3 cipher suites are currently considered safe, no additional configuration is required for security considerations for this protocol.

TLS 1.2 is different. There are many different TLS cipher suites available with Apache version 2.4.55,⁷ which can be viewed with the command

```
openssl ciphers -v
```

By default, only two rules are defined regarding ciphers:

- 1) HIGH – some ciphers with a key length of 128 bits and all stronger ones are enabled;
- 2) !aNULL – ciphers not supporting authentication are disabled.

```
SSLCipherSuite HIGH:!aNULL
```

Picture 30 – server-based configuration in the file `/etc/apache2/mods-available/ssl.conf`

If you wish to configure the available cipher suites used with TLS 1.2 in more detail, you can use the command 'SSLCIPHERSUITE' in the Apache configuration file. Here, you can use predefined aliases or exact cipher suite descriptions.

⁷ The ciphers of other protocols are not covered in this chapter, because protocols older than version 1.2 should be disabled and version 1.3. is currently preferred.



It is impossible to give an exact recommendation for configuring cipher suites without knowing the requirements applicable to the webpage. However, non-secure cipher suites must be removed from the list. It is reasonable to describe the specific enabled cipher suites for TLS 1.2.

Example:

- Using the following command line in the configuration file:
SSLCIPHERSUITE "ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384"
only the described cipher suites are allowed.

You can also configure cipher suites at the server level by modifying the parameter 'SSLCIPHERSUITE' in the file /etc/apache2/mods-available/ssl.conf.

More information about the recommendations for the use of the cipher suites can be found in the cryptographic algorithms life cycle report ordered by RIA at <https://www.id.ee/en/article/cryptographic-algorithms-life-cycle-reports-2/>.

SSLHONORCIPHERORDER

Another important parameter related to ciphers is 'SSLHONORCIPHERORDER', the value of which should be set to ON in the configuration file. This way, the server's list of cipher suites is always preferred over the user's. By default, this parameter is undefined and its default value is 'off'.

Additional filtering of user certificates

Important! To avoid access to the web service with incorrect certificates, you must add the following requirements in the server configuration:

- 1) The certificate must have the correct 'extendedKeyUsage' field;
- 2) The issuer of the certificate must be ESTEID2018 or ESTEID2025.

Add the following lines to the Apache configuration:

```
<Location "/">
Require expr ( \
    %{SSL_CLIENT_I_DN_CN} == "ESTEID2018" || %{SSL_CLIENT_I_DN_CN} ==
"ESTEID2025") \
and "TLS Web Client Authentication, E-mail Protection" in
PeerExtList('extendedKeyUsage') \
)
</Location>
```

The configuration above can be added to the virtual host or to the Apache main configuration. After adding these requirements, only certificates with the correct *extendedKeyUsage* field issued from the enabled chain can access the services.



Notes:

- If you are using another feature to filter network traffic, the secure configuration should be implemented there, too. SK has published information about the F5 configuration in the chapter 'Only accept certificates with trusted key usage' in the following article: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>
- SK's recommendations for secure ID-card authentication are published here in the chapter 'Defence: implement ID-card authentication securely': <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>
- The recommended method for avoiding incorrect certificates is using OIDs in the certificates. Unfortunately, there is currently no method for doing this at the server level. If possible, open the certificate at the web application level, check for a correct OID in the certificate, and if there is none, reject the authentication request. All currently known OIDs are listed in the chapter 'Only accept certificates with trusted issuance policy' in the following article published by SK: <https://github.com/SK-EID/smart-id-documentation/wiki/Secure-Implementation-Guide>

Filtering certificates displayed to the user

By default, the selection of certificates displayed to the user is not limited, which means that all user authentication certificates are listed during authentication in the web server. However, you should only display the certificates issued from the ESTEID2018 or ESTEID2025 chain to the user. To do so:

- 1) Create a file for accepted chains **/etc/ssl/certs/DN_Bundle.pem⁸**
- 2) Put the ESTEID2018 and ESTEID2025 certificates in Base64 format into the created file
- 3) add the directive
SSLCADNRequestFile /etc/ssl/certs/DN_Bundle.pem
into the SSL section of the Apache configuration file and save the new configuration;
- 4) restart the Apache server with
systemctl reload apache2

Now, Apache will send information to the user that only certificates issued from the ESTEID2018 chain are supported and only certificates issued from that chain are displayed to the user.

Enabling HTTP Strict Transport Security (HSTS)

1. Enable mod-headers in the terminal with
a2enmod headers

⁸ Available as DN_Bundle.pem

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards

```
root@ubuntu2204: ~
root@ubuntu2204:~# a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  systemctl restart apache2
root@ubuntu2204:~# systemctl restart apache2
root@ubuntu2204:~#
```

Picture 31 – enabling Apache headers (in this example, for HSTS)

2. Add the following line to the Apache configuration:
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"

```
# Enable HSTS.
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
```

Picture 32 – activating HSTS for 30 days

3. To apply the changes, restart the Apache service.

Additional possibilities

In addition to TLS and cipher suite configuration, you should pay attention to the security of the Apache server from the following aspects:

- Keep the operating system up to date.
- Keep Apache up to date.
- Run Apache under non-root user rights.
- Disable presenting server information.
- Remove unnecessary modules.
- Add and configure *Mod Security*.
- Add and configure *Mod Evasive*.
- Disable default directory listing.
- Enable logging.
- ...

The above is a sample list of ways to improve Apache security. Detailed recommendations are available online: <https://www.google.com/search?q=how+to+secure+apache+server>.

Ubuntu/Apache2 SSL configuration



A simple configuration guide for Estonian ID-cards

Appendix

EID_Bundle.pem

```
EE-GovCA2018
-----BEGIN CERTIFICATE-----
MIIE+DCCBFmgAwIBAgIQMLOWlXoR0oFbj52nmRsneZAKBggqhkJOPQQDBDBaMQsw
CQYDVQQGEwJFRTEbMBkGA1UECgwSU0sgSUQgU29sdXRpb25zIEFTMRcwFQYDVQRh
DA5OVFJFRS0xMDC0NzAxMzEVMBMGA1UEAwwMRUUtR292Q0EyMDE4MB4XDTE4MDkw
NTA5MTEwM1oXDTMzMDkwNTA5MTEwM1owWjELMAkGA1UEBhMCRUUXGzAZBgNVBAoM
E1NLEIEIEIFNvbHV0aW9ucyBBUzEXMBUGA1UEYQwOT1RSRUUtMTA3NDcwMTMxFTAT
BgNVBAMMDEVFLUdvdnkNBMjAxODCBmzAQBgqhkJOPQIBBjUrgQQAiw0BhgAEAMcb
/dmAcVo/b2azEPS6CFW7fEA2KuHKC53D7ShVNVLz4QUjCdTXjds/4u99jUoYEQec
luVVzmlgEJR1nkN2eOrLAZYxPjwG5Hi1liZEyW9QKVdeEgyvzhWWTNHGjV3HdZRv
7L9o4533PtJAYqJ90aTIB9zCCAfmwCAYGBACPEgECMAkGBwQAI+xAAQIwMgYLKwYB
BAGDkSEBAQEwIzAhBggrBgEFBQCARYVaHR0cHM6Ly93d3cuc2suZWUvQ1BTMA0G
CysGAQQBg5EhAQECMA0GCysGAQQBg5F/AQEEMA0GCysGAQQBg5EhAQEFMA0GCysG
AQQBg5EhAQEGMA0GCysGAQQBg5EhAQEHMA0GCysGAQQBg5EhAQEDMA0GCysGAQQB
g5EhAQEEMA0GCysGAQQBg5F/AQEEMA0GCysGAQQBg5EhAQEJMA0GCysGAQQBg5Eh
AQEKMA0GCysGAQQBg5EhAQELMA0GCysGAQQBg5EhAQEMMA0GCysGAQQBg5EhAQEN
MA0GCysGAQQBg5EhAQEOMA0GCysGAQQBg5EhAQEPMA0GCysGAQQBg5EhAQEQMA0G
CysGAQQBg5EhAQERMA0GCysGAQQBg5EhAQESMA0GCysGAQQBg5EhAQETMA0GCysG
AQQBg5EhAQEUAMA0GCysGAQQBg5F/AQEEMA0GCysGAQQBg5F/AQEDMA0GCysGAQQB
g5F/AQEEMA0GCysGAQQBg5F/AQEFMA0GCysGAQQBg5F/AQEGMDEGCisGAQQBg5Eh
CgEwIzAhBggrBgEFBQCARYVaHR0cHM6Ly93d3cuc2suZWUvQ1BTMBGCCSQAQUF
BwEDBAwwCjAIBgYEAISGAQEwCgYIKoZIzj0EAwQDgYwAMIGIAkIBk698EquetY9Tt
6Hw050CfzdIjKmlfCI34xkdU7J+wz1tNVu2tHJwEhdsH0e92i969sRDp1RNP1Vh
4XFJzI3oQFQCQgVxmcuVnsy7NUScDZ0erwovmbFOsNxELCANxNSWx5xMqzEiHv8
46opxu10UFDIBBPzkbBenL4h+g/WU71G78fIhA==
-----END CERTIFICATE-----
ESTEID2018
-----BEGIN CERTIFICATE-----
MIIFVzCCBLigAwIBAgIQdUf6rBR0S4tbo2bU/mZV7TAKBggqhkJOPQQDBDBaMQsw
CQYDVQQGEwJFRTEbMBkGA1UECgwSU0sgSUQgU29sdXRpb25zIEFTMRcwFQYDVQRh
DA5OVFJFRS0xMDC0NzAxMzEVMBMGA1UEAwwMRUUtR292Q0EyMDE4MB4XDTE4MDkw
NTA5MTEwM1oXDTMzMDkwNTA5MTEwM1owWDELMAkGA1UEBhMCRUUXGzAZBgNVBAoM
E1NLEIEIEIFNvbHV0aW9ucyBBUzEXMBUGA1UEYQwOT1RSRUUtMTA3NDcwMTMxFTAT
BgNVBAMMCKVTVEVJRDIwMTgwgZswEAYHkoZiZj0CAQYFK4EEACMDgYYABAHHOB1v
7URPYP1yHhOb7RA/YBDbtgygnSVmQYdxnFrKHUXh6tFkghvHuA1k2DSom1hE5kqh
B5VspDembwWDJBOQWGOI/0t3EtcCLyJeM7F9xOPdzUbZaIbpNRHpQgVbPFX0xpL
TgW27MpImH08DHBWfPeAaNX3eUpD4gC5cvhsK0RFEqOCAx0wggMZMB8GA1UdIwQY
MBaAFH4pVuc0knhOd+FvLjMqmHHB/TSfMB0GA1UdDgQWBbTzrHDbx36+1Pig5L5H
otA0rZoaqEjAObgNVHQ8BAf8EBAMCAQYwEgYDVR0TAQH/BAgwBgEB/wIBADCCAc0G
A1UdIASCAcQwggHAMAGBgQAj3oBAjAJBgCEAIVsQAECMDIGCysGAQQBg5EhAQEB
MCMwIQYIKwYBBQUHAgEWFwh0dHBz0i8vd3d3LnNrLmV1L0NQUzANBgsrBgEEAYOR
IQEBjANBgsrBgEEAYORfweBATANBgsrBgEEAYORIQEBBTANBgsrBgEEAYORIQEB
BjANBgsrBgEEAYORIQEBBzANBgsrBgEEAYORIQEBBzANBgsrBgEEAYORIQEBBDAN
BgsrBgEEAYORIQEBBDANBgsrBgEEAYORIQEBCTANBgsrBgEEAYORIQEBcANBgsr
BgEEAYORIQEBcANBgsrBgEEAYORIQEBDDANBgsrBgEEAYORIQEBDTANBgsrBgEE
AYORIQEBDjANBgsrBgEEAYORIQEBDzANBgsrBgEEAYORIQEBEDANBgsrBgEEAYOR
IQEBETANBgsrBgEEAYORIQEBEjANBgsrBgEEAYORIQEBEzANBgsrBgEEAYORIQEB
FDANBgsrBgEEAYORfweBAjANBgsrBgEEAYORfweBAzANBgsrBgEEAYORfweBBAN
BgsrBgEEAYORfweBBTANBgsrBgEEAYORfweBBjAqBgNVHSUBAf8EIDAeBggrBgEF
BQCDCQYIKwYBBQUHAgIGCCSQAQUFBwMEMGoGCCSQAQUFBwEBBF4wXDAPBggrBgEF
BQCwAYYdaHR0cDovL2FpYzZyYzY1ZS91ZS1nb3ZjYTYwMTgwgLwYIKwYBBQUHMAK
I2h0dHA6Ly9jLnNrLmV1L0VFLUdvdnkNBMjAxOC5kZXIuY3J0MBGgCCSQAQUFBwED
BAwwCjAIBgYEAISGAQEwMAYDVR0fBCKwJzAlc0CgIYYfaHR0cDovL2Muc2suZWUv
RUUtR292Q0EyMDE4LmNybDAKBggqhkJOPQQDBAObJAawgYgCQgDeUUY4HczUbFKS
002HZ88gclgYdzthGqLENYTmtXE6dMBRnCbGUmhBCAAOmJSHbyFJ8W99ikLiSyurm
kJM0hDE9KgJCASOq405Ia5nKjTJPNsHQ1Mi7KZsIcTH0oBccx+54N8ZX1MgBozJ
mT59rZy/2/OeE163BAwD0UduQUAnMPP6+W3Vd
-----END CERTIFICATE-----
EEGovCA2025
```